

Tablice

Podstawy programowania

Czym są tablice?



Tworzenie tablic

typ[] nazwaTablicy

Np.

int[] liczby;

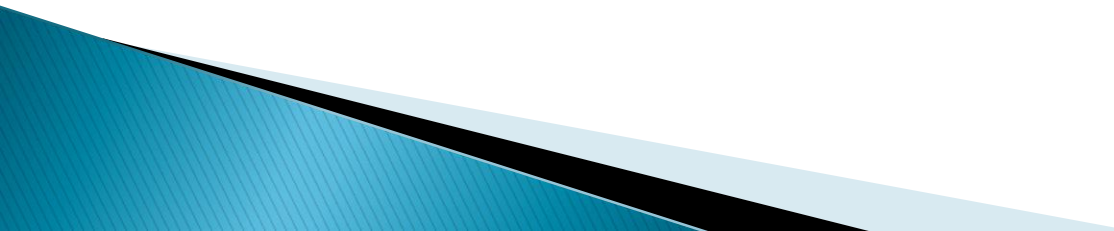
Tworzenie tablicy:

liczby = new int[10];

Lub

int [] nazwaTablicy = new int[10];

*double[] pomiary = {2.1, 4.0, 3.1, 3.0, 2.8, 3.2,
2.7, 3.8, 2.5, 2.9};*



Odwołanie do tablicy

tablica[indeks]

np.

suma = tab[1] + tab[2];

suma += tab[n];

tab.length – zwraca liczbę elementów tablicy

Kopiowanie tablic

```
int[] tablicaOryginalna = {1, 2, 3, 4};  
int[] tablicaSkopiowana = tablicaOryginalna;  
//co się stanie??
```

Kopiowanie tablic

- ▶ Sposób 1:

```
int[] oryginal = {1, 2, 3, 4};  
int[] kopia = new int[4];  
for (int i = 0; i < oryginal.length; i++) {  
    kopia[i] = oryginal[i];  
}
```

- ▶ Sposób 2:

```
int[] kopia = (int[])oryginal.clone();
```

Tablice dwuwymiarowe

int[][] tablica2D;

tablica2D = new int[10][10];

Lub

int[][] tablica2D = new int[10][10];

Wypełnienie elementami:

int[][] tablica2 = {{1,2,3,4},{5,6,7,8},{9,10,11,12}};

Tablice nieprostokątne

```
// tablica4 będzie przechowywać 4 tablice jednowymiarowe,  
// na razie nie jest określone jakiej będą długości  
int[][] tablica4 = new int[4][];
```

```
// tablica4 w pierwszej komórce będzie przechowywać 5-elementową  
// tablicę  
tablica4[0] = new int[5];
```

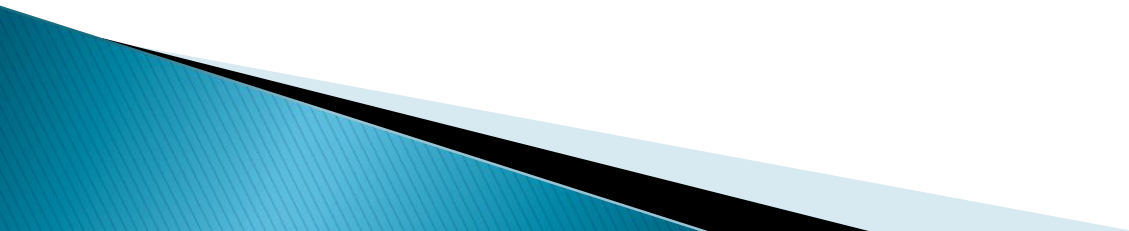
```
// tablica 4 będzie przechowywać komórce pod indeksem 3  
// tablicę o 3 elementach  
tablica4[3] = new int[3];
```

```
// umieszczenie elementu  
tablica4[3][2] = 2;
```

```
// pobranie elementu  
▶ System.out.println("tablica4[3][2]: "+tablica4[3][2]);
```


Zadanie 1

Zadeklaruj tablicę dwuwymiarową typu `int` i oblicz sumę elementów.

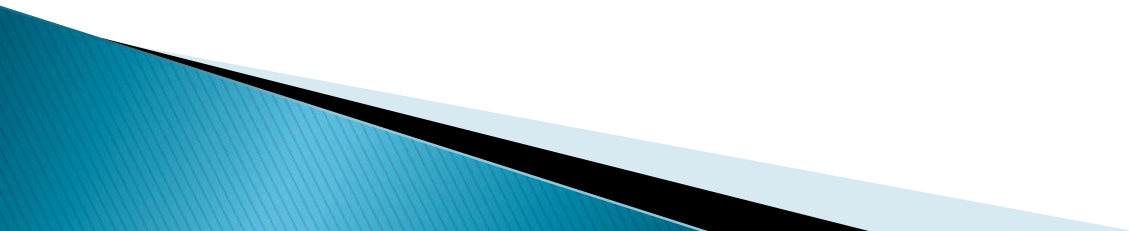


Zadanie 2

Utwórz tablicę 10x5, wypełnij ją losowymi liczbami całkowitymi (0–20), a następnie sprawdź w którym wierszu jest największa suma (wypisz liczby na ekran, obok podaj sumę każdego wiersza).

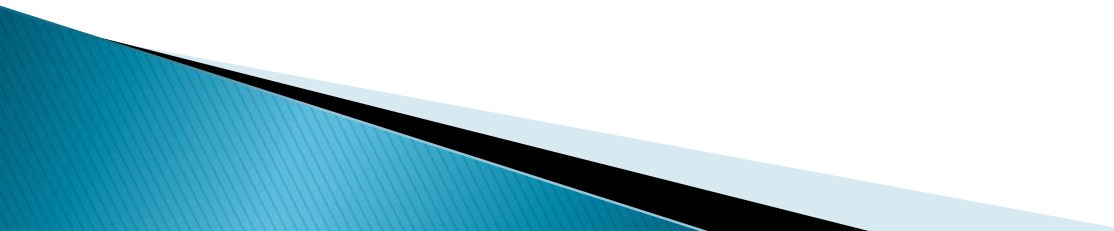
Zadanie 3

Napisz program sortujący tablicę 20 elementową. (sortowanie bąbelkowe)



Zadanie 4

Utwórz dwie tablice dwuwymiarowe 4x4.
Wypełnij je losowymi danymi typu int (0–100). Utwórz trzecią tablicę i wypełnij ją tak, że element na pozycji [i,j] będzie sumą odpowiednich elementów z dwóch poprzednich tablic (suma macierzowa).



Zadanie 5

Użytkownik podaje 10 liczb z klawiatury ($\text{int} > 0$). Wczytuj te liczby do kolejnych wierszy tablicy `dane[]`, tak aby na pierwszym miejscu była wczytywana liczba (n), a następnie n liczb losowych z zakresu $(0; 2n]$

Zadanie 6

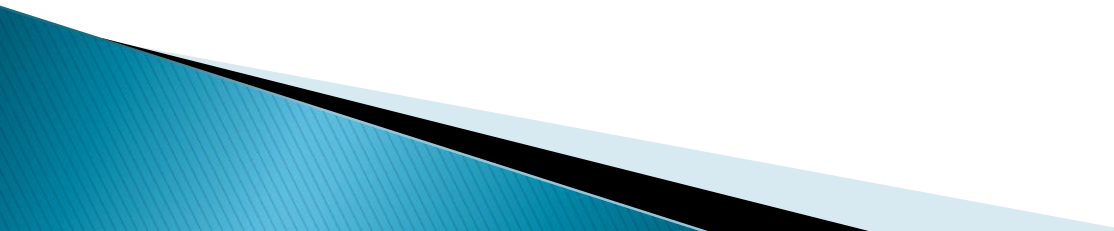
Napisać program, który:

- ▶ utworzy tablicę 10 liczb całkowitych i wypełni ją wartościami losowymi z przedziału $[-10, 10]$,
- ▶ wypisze na ekranie zawartość tablicy,
- ▶ wyznaczy najmniejszy oraz największy element w tablicy,
- ▶ wyznaczy średnią arytmetyczną elementów tablicy,
- ▶ wyznaczy ile elementów jest mniejszych, ile większych od średniej.
- ▶ wypisze na ekranie zawartość tablicy w odwrotnej kolejności, tj. od ostatniego do pierwszego.

Wszystkie wyznaczone wartości powinny zostać wyświetlone na ekranie.

Zadanie 7

Napisać program, który utworzy tablicę 20 liczb całkowitych z przedziału 1 . . . 10, a następnie wypisze na ekranie ile razy każda z liczb z tego przedziału powtarza się w tablicy.

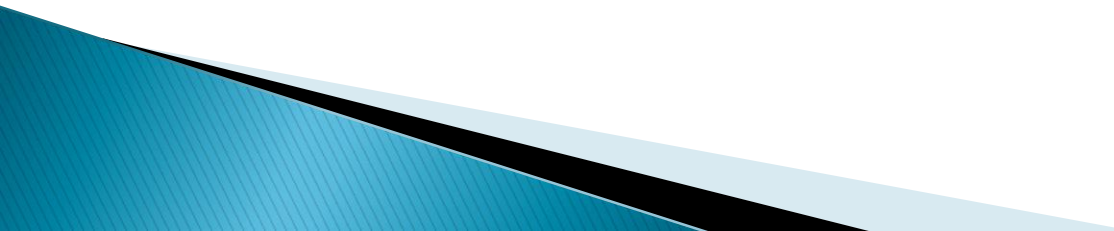


Zadanie 8

Napisać program, który wczytuje od użytkownika liczbę całkowitą, a następnie wyświetla jej reprezentację w kodzie binarnym (ZM).

Podczas konwersji liczby należy kolejne jej bity zapisywać w pomocniczej tablicy liczb całk. o rozmiarze 32.

Konwersji należy dokonać korzystając z operacji dzielenia całkowitego oraz operacji modulo.



Zadanie 9

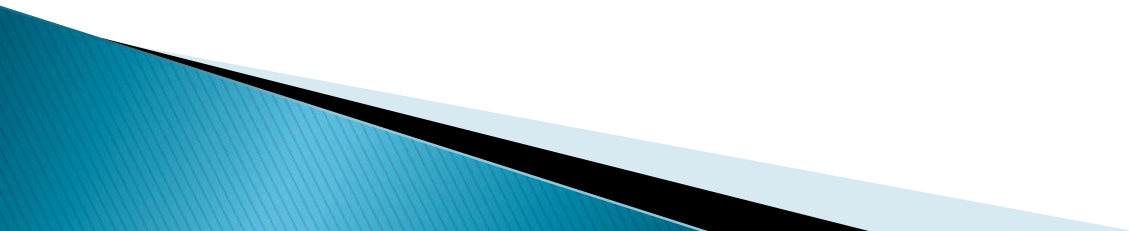
Zadeklaruj tablicę o rozmiarze 50. Wypełnij tablicę zgodnie z regułami Ciągu Fibbonacciego (pierwszy i drugi element = 1, każdy następny to suma dwóch poprzednich: 1, 1, 2, 3, 5, 8);

Zadanie 10

Zadeklaruj tablicę String, podaj n słów, a następnie wypisz z tablicy słowa kończące się na "-a" lub zawierające literę "-b".

Zadanie *

Napisz program realizujący grę sudoku



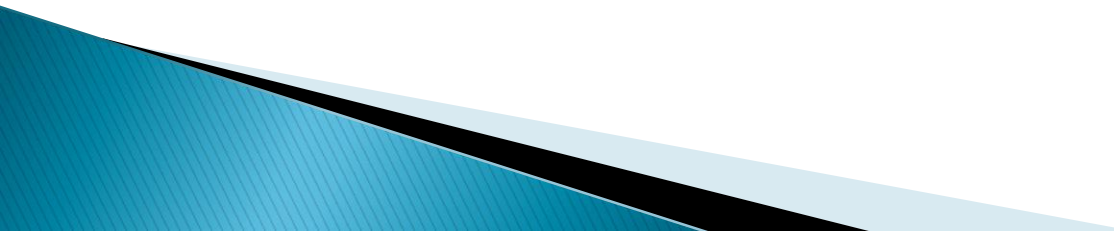
Operacje na plikach, wyjątki



Odczyt pliku

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Odczyt{
    public static void main(String[] args) throws FileNotFoundException{
        File file = new File(„plik.txt");
        Scanner in = new Scanner(file);
        String zdanie;
        while (true){
            zdanie = in.nextLine();
            if (zdanie == null) break;
            System.out.println(zdanie);
        }
    }
}
```



Zapis do pliku

```
import java.io.FileNotFoundException;
```

```
import java.io.PrintWriter;
```

```
public class Zapis{
```

```
    public static void main(String[] args) throws  
        FileNotFoundException{
```

```
        PrintWriter zapis = new PrintWriter(„dokument.txt”);
```

```
        zapis.println(„Jakies przykładowe zdanie”);
```

```
        zapis.close();
```

```
    }
```

```
}
```



Wyjątki

```
try{
```

```
    kod programu mogący generować wyjątki
```

```
}
```

```
catch (TypWyjątku1 a){ Obsługa wyjątku a }
```

```
catch (TypWyjątku2 b){ Obsługa wyjątku b }
```

```
...
```

```
finally{ Blok instrukcji, który wykona się  
    niezależnie, czy wyjątki wystąpią, czy nie }
```

Wyjątki przykład

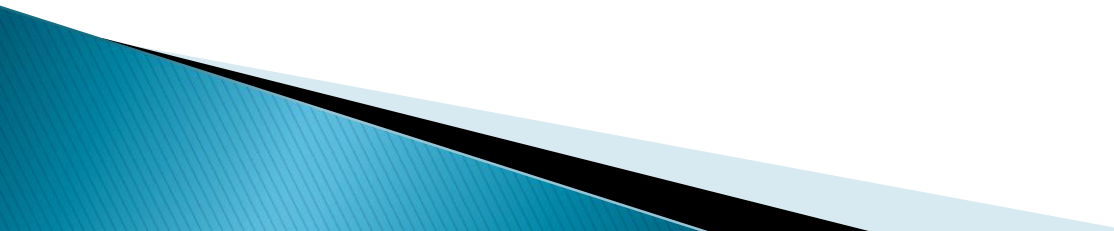
```
public int liczbaSekund(int godziny)
{
    if (godziny < 0) {
        throw new IllegalArgumentException(„Godzina musi być >= 0: "
        + godziny); }
    return godziny * 60 * 60; }
```


Wyjątki przykład

```
int gdz = -3;  
int lSek = 0;  
try {  
    lSek = klasa. liczbaSekund(gdz);  
}  
catch (IllegalArgumentException exception) {  
    lSek = klasa.liczbaSekund(gdz * -1);  
}  
System.out.println(lSek);
```

Wyjątki przykład

```
try {  
    index = Integer.parseInt(odczyt.readLine());  
}  
catch (NumberFormatException n) {  
    System.out.println("Niepoprawne dane! " +  
        "\n Który element tablicy chcesz zobaczyć: ");  
}  
catch (IOException e) { System.out.println("Błąd odczytu  
danych");  
}
```



Zadanie 1

Napisać funkcję `liczZnakiSlowa`, która zlicza:

- ▶ liczbę znaków w pliku,
- ▶ liczbę białych znaków w pliku (białe znaki to spacja, tabulator, znacznik końca wiersza),
- ▶ liczbę słów w pliku.

Wynikiem funkcji jest tablica złożona z 3 liczb całkowitych po jednej dla wymienionych podpunktów.

Zadanie 2

Napisać funkcję:

```
public static void szukaj(String nazwaPlikWe, String nazwaPlikWy, String slowo)
```

której zadaniem jest znalezienie wszystkich wierszy w pliku, które zawierają szukane słowo.

Wszystkie wiersze, które zawierają słowo powinny zostać zapisane w pliku wynikowym wraz z nr wiersza (z pierwszego pliku).

Nazwa pierwszego pliku zapamiętana jest w parametrze nazwaPlikWe, nazwa pliku wynikowego podana jest w parametrze nazwaPlikWy, natomiast szukane słowo w parametrze slowo.

Przykład – plik wejściowy:

Ala ma jutro egzamin z biologii.

Jan myje auto.

Eh, jutro kolejny egzamin.

Nie lubię polityki.

Jeżeli szukany słowem byłoby "egzamin", to plik wynikowy powinien wyglądać następująco:

1: Ala ma jutro egzamin z biologii.

3: Eh, jutro kolejny egzamin.

Zadanie 3

Napisać funkcję

public static void sumujIZapisz(String nazwaPliku)

która odczytuje plik o podanej nazwie zawierający liczby całkowite (po jednej w wierszu).

Funkcja ma za zadanie odczytać i zsumować wszystkie liczby z pliku, a następnie dopisać na końcu pliku wyznaczoną sumę powiększoną o 1. Ponowne uruchomienia funkcji będą skutkowały dopisywaniem kolejnych wierszy.

Jeżeli plik nie istnieje to ma zostać utworzony – suma dla pustego pliku wyniesie 0, a więc należy dopisać wiersz zawierający 1.

Zadanie 4

Napisać funkcję

emerytura(String nazwaPliku)

która wczyta z pliku o podanej nazwie dane pracowników zapisane w kolejnych wierszach w następujący sposób:

Imię Nazwisko Płeć Wiek

Przykład:

Tomasz Nowak M 45

Marta Ziobro K 42

Jan Kowalski M 27

Ewelina Tusk K 59

Następnie funkcja dla każdego pracownika powinna wyznaczyć ile lat pozostało do jego emerytury.

Wyniki należy zapisać w następujący sposób:

Nazwisko Imię "Lata do emerytury"

Przykład:

Nowak Tomasz 20

Kowalski Jan 38

Wyniki dla kobiet należy zapisać w pliku o nazwie "kobiety.txt", natomiast wyniki dla mężczyzn należy zapisać w pliku "mężczyzni.txt"

Zadanie 5

Napisać funkcję, której zadaniem jest odczytanie danych tabelarycznych z pliku tekstowego, a następnie zapisanie ich do nowego pliku w postaci kodu HTML.

Przykład:

Wejście:

"Waga" "Wzrost" "BMI" "Nadwaga"

70 1,8 21,6 "NIE"

67 1,77 21,39 "NIE"

85 1,7 29,41 "TAK"

100 1,92 27,13 "TAK"

Wynik:

Wynik:

```
<html><body>
```

```
<table>
```

```
<tr><td>"Waga"</td><td>"Wzrost"</td><td>"BMI"</td><td>"Nadwaga"
```

```
</td></tr>
```

```
<tr><td>70</td><td>1,8</td><td>21,6</td><td>"NIE"
```

```
</td></tr>
```

```
<tr><td>67</td><td>1,77</td><td>21,39</td><td>"NIE"
```

```
</td></tr>
```

```
<tr><td>85</td><td>1,7</td><td>29,41</td><td>"TAK"
```

```
</td></tr>
```

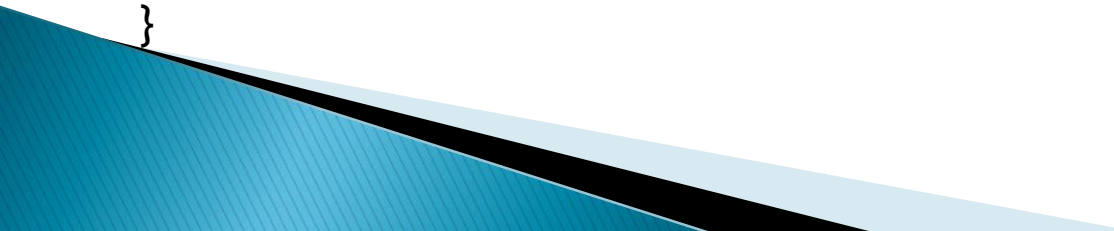
```
<tr><td>100</td><td>1,92</td><td>27,13</td><td>"TAK"</td></tr>
```

```
</table>
```

```
</body></html>
```

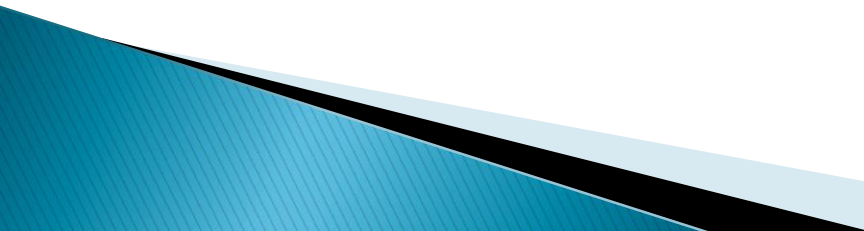
Pliki tekstowe – odczyt

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class Odczyt{
    public static void main(String[] args) throws FileNotFoundException{
        File file = new File(„plik.txt");
        Scanner in = new Scanner(file);
        String zdanie;
        while (zdanie = in.nextLine() != null) {
            System.out.println(zdanie);
        }
        in.close();
    }
}
```



Pliki tekstowe – zapis

```
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
public class Zapis{  
    public static void main(String[] args) throws FileNotFoundException{  
        PrintWriter zapis = new PrintWriter(„dokument.txt”);  
        zapis.println(„Jakies przykładowe zdanie”);  
        zapis.close();  
    }  
}
```



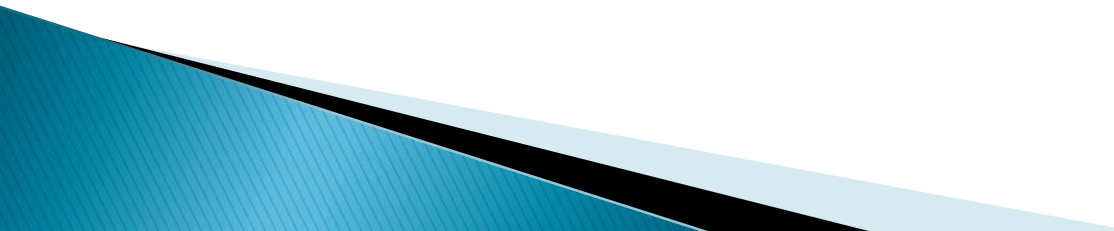
Pliki o dostępie swobodnym

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;
public class Pliki_RAF {
    public static void main(String[] args) {
        String tekst="";
        long pozycja=0, dlugoscpliku=0;
        RandomAccessFile raf = null;
        // OTWIERANIE PLIKU I WYBÓR TRYBU ODCZYTU - "r" (zapis - "w", zapis/odczyt "rw")
        try {
            raf = new RandomAccessFile("plik.txt", "r");
            dlugoscpliku = raf.length();
            System.out.println("Dlugosc pliku: " + dlugoscpliku);
            raf.seek(0); // przejście na początek pliku
            while (pozycja < dlugoscpliku){
                tekst += raf.readLine(); //readInt() readDouble() ...
                pozycja= raf.getFilePointer();
            }
        } catch (FileNotFoundException e) {
            System.out.println("Błąd otwarcia");
        }
    }
}
```

Do zapisu do pliku writeInt(), writeDouble(), writeLine()...

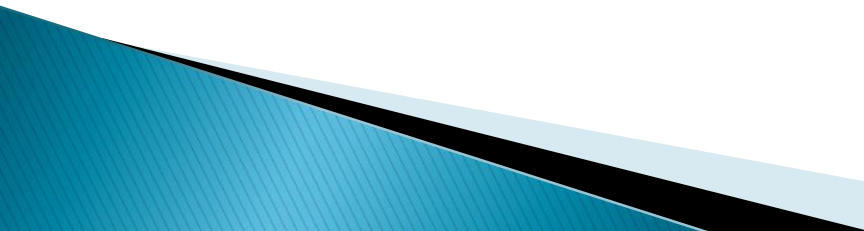
Bliki binarne – odczyt

```
String filePath = „plik.txt”
int number = 0;
BufferedReader fileReader = null;
try {
    fileReader = new BufferedReader(new FileReader(filePath));
    String numberAsString = fileReader.readLine();
    number = Integer.parseInt(numberAsString);
} catch(IOException e){
    System.out.println(„Błąd odczytu”);
}finally {
    if (fileReader != null) {
        fileReader.close();
    }
}
```



Pliki binarne – zapis

```
String filePath = „plik.txt”
int number = 1234567;
DataOutputStream outputStream = null;
try {
    outputStream = new DataOutputStream(new
        FileOutputStream(filePath));
    outputStream.writeInt(number);
} catch (IOException e) {
    System.out.println(„Błąd odczytu”);
} finally {
    if (outputStream != null) {
        outputStream.close();
    }
}
```



Zadania

Zad. B2.

Napisać program przetwarzający plik zawierający łańcuchy znaków reprezentujące wyrażenia zapisane w postaci postfiksowej ($ab+$) ze zmiennymi a i b oraz operatorami 2-argumentowymi: $+$, $-$, $*$, $/$. W pliku, za każdym łańcuchem, jest umieszczona liczba całkowita reprezentująca informację o poprawności wyrażenia (na początku równa -1). Program powinien uzupełnić w pliku informacje o poprawności wyrażen, wpisując 0 , gdy wyrażenie jest niepoprawne i 1 , gdy wyrażenie jest poprawne.

Przykładowe wyrażenie poprawne:

$a\ abba+-* \ ab*a+ba- /$

i niepoprawne:

$abc+$ (za mało operatorów)

$ab+a-+$ (za mało argumentów)

Zadania

Napisać funkcję:

`double liczbaD(String łańcuch)`

przekształcającą łańcuch binarny reprezentujący wewnętrzną postać wartości typu `double` na liczbę, działając według następującego algorytmu:

1. Wylicz znak liczby:

bit 0: 0 → dodatnia, 1 → ujemna

2. Wylicz cechę (bity 1–11) i odejmij 127

3. wylicz mantysę (bity 12–63) jako ułamek i dodaj 1

4. Wylicz wartość liczby jako:

$\pm \text{mantysa} \cdot 2^{\text{cecha}}$

Zastosować metody:

- `charAt(nrZnaku)` klasy `String` (nie stosować metody `substring()`)
- `parseLong(łańcuch, podstLiczenia)` klasy `Long`
- `pow(liczba, wykładnik)` klasy `Math`

Zadania

Zad. B1.

Napisać funkcję Emerytura(plik), która na podstawie pliku o podanej nazwie zawierającego dane pracowników zapisane w kolejnych wierszach w następujący sposób:

„Imię-Nazwisko-Płeć-Wiek”,

dla każdego pracownika powinna wyznaczyć ile miesięcy pozostało mu do emerytury (kobiety do 65, a mężczyźni – do 67 roku życia). Wyniki należy zapisać następująco:

Nazwisko

Miesiące

Wyniki dla kobiet należy zapisać w pliku danych o nazwie „kobiety.bin”, natomiast wyniki dla mężczyzn należy zapisać w pliku „meczczyni.bin”.

Przykład pliku wejściowego:

Tomasz-Nowak-M-45

Marta-Głowa-K-42

Jan-Kowalski-M-27

Ewelina-Tabor-K-59

Przykład dla mężczyzn:

Nowak

264

Kowalski

348